

In the claims:

For the Examiner's convenience, all pending claims are presented below with changes shown.

1. (Currently Amended) A method comprising:  
receiving a string of data symbols; and  
compressing the string of data into a fixed sized compressed data block having a plurality of compressed symbols and dictionary elements, the compressed symbols and dictionary elements having a fixed length and a fixed offset within the compressed data block, wherein the number of dictionary elements is automatically derived from a number of leading bits in the string of data ~~each of the compressed symbols include encoded tag bits to indicate a type of compression performed on the associated compressed symbol.~~
2. (Original) The method of claim 1 wherein compressing the data comprises:  
dividing a first symbol into a first component and a second component; and  
comparing the first component with the dictionary elements.
3. (Original) The method of claim 2 further comprising compressing the first component to form a first tag if the first component matches a dictionary element.
4. (Cancelled)
5. (Original) The method of claim 3 further comprising storing the first component at a dictionary element if the first component does not match a dictionary element.
6. (Currently Amended) The method of claim 3 wherein compressing the data comprises:

dividing a second symbol into a ~~second~~ first component and a second component;  
and  
comparing the second component with the dictionary elements.

7. (Currently Amended) A compression system:

a register to store a plurality of fixed length data symbols to be compressed;  
compression logic to compress each of the plurality of data symbols to form a  
compressed symbol; and  
a plurality of dictionary registers to store dictionary elements,  
the compressed symbols and dictionary elements forming a compressed data block  
having a fixed length and fixed offset, wherein the number of dictionary elements is  
automatically derived from a number of leading bits in the string of data each of the  
compressed symbols include encoded tag bits to indicate a type of compression performed on  
the associated compressed symbol.

8. (Original) The system of claim 7 wherein each symbol is divided into a first  
component and a second component.

9. (Previously Presented) The method of claim 8 wherein the first and second  
components are compressed into fixed length compressed symbols.

10. (Cancelled)

11. (Original) The system of claim 8 wherein the first component is received at the  
compression logic and encoded to form a tag.

12. (Original) The system of claim 11 further comprising a buffer to store the tag and second component of each symbol as the compressed symbol.

13. (Previously Presented) The system of claim 8 wherein the compression logic comprises:

dictionary matching logic to determine if the first component matches a dictionary element; and

no match logic to determine if the second component has all ones or all zeroes.

14. (Original) The system of claim 13 wherein the compression logic comprises an encoder coupled to the match logic and the no match logic to encode the first component to form a tag if the first component matches a dictionary element, has all ones or zeroes.

15. (Previously Presented) A method comprising:

receiving a fixed offset compressed data block having a plurality of dictionary elements and compressed symbols; and

decompressing each of the compressed symbols in parallel, by:

analyzing encoded tag bits within a compressed symbol; and

decompressing the compressed symbol to form a symbol based upon a type of compression indicated by the encoded tag bits, wherein each of the compressed symbols are decompressed simultaneously.

16. (Cancelled)

17. (Cancelled)

18. (Previously Presented) The method of claim 15 wherein decompressing the compressed symbol to form a symbol based upon the tag value comprises:

decoding the tag to form a matched component of the symbol; and

combining the matched component with an unmatched component within the compressed symbol to form the symbol.

19. (Previously Presented) A decompression system comprising:

a plurality of decompression units to decompress a corresponding compressed symbol within a compressed data block having a plurality of compressed symbols and dictionary elements having a fixed length and fixed offset to generate an uncompressed symbol by analyzing encoded tag bits within a compressed symbol and decompressing the compressed symbol to form a symbol based upon a type of compression indicated by the encoded tag bits, wherein the decompression units decompress the compressed symbols in parallel.

20. (Cancelled)

21. (Original) The system of claim 19 wherein each decompression unit comprises logic to decode the encoded tag bits of a compressed symbol to generate a matched symbol component.

22. (Original) The system of claim 21 wherein each decompression unit combines a matched symbol component with the unmatched symbol component to form an uncompressed symbol.

23. (Currently Amended) A computer system comprising:  
a central processing unit (CPU);

a cache memory coupled to the CPU having a plurality of compressible cache lines to store additional data; and

a cache controller comprising compression logic to compress each of the plurality of cache lines by compressing the data within a compressed cache line into a fixed sized compressed data block having a plurality of offset compressed symbols and dictionary elements, the symbols and dictionary elements having a fixed length and fixed offset, wherein the number of dictionary elements is automatically derived from a number of leading bits in the string of data ~~each of the compressed symbols include encoded tag bits to indicate a type of compression performed on the associated compressed symbol.~~

24. (Original) The computer system of claim 23 wherein the cache controller further comprises decompression logic to decompress compressed symbols within a compressed data block to generate uncompressed symbols.

25. (Original) The computer system of claim 24 wherein the decompression logic decompresses the compressed symbols in parallel.

26. (Currently Amended) A computer system comprising:

a central processing unit (CPU);

a cache memory coupled to the CPU having a plurality of compressible cache lines to store additional data;

a chipset, coupled to the CPU and the cache memory, including:

compression logic to compress each of the plurality of cache lines by compressing the data within a compressed cache line into a fixed sized compressed data block having a plurality of offset compressed symbols and dictionary elements,

the symbols and dictionary elements having a fixed length and fixed offset, wherein the number of dictionary elements is automatically derived from a number of leading bits in the string of data ~~each of the compressed symbols include encoded tag bits to indicate a type of compression performed on the associated compressed symbol;~~ and a main memory coupled to the chipset;

27. (Original) The computer system of claim 26 wherein the chipset further comprises decompression logic to decompress compressed symbols within a compressed data block to generate uncompressed symbols.

28. (Previously Presented) A method comprising:  
receiving a fixed offset compressed data block having a plurality of dictionary elements and compressed symbols; and  
decompressing a randomly accessed and a first compressed symbol within the compressed data block by:  
analyzing encoded tag bits within a compressed symbol; and  
decompressing the compressed symbol to form a symbol based upon a type of compression indicated by the encoded tag bits.

29. (Cancelled)